Towards Adaptive Optimal Control of the Scramjet Inlet

Nilesh V. Kulkarni

Advisors:

Prof. Minh Q. Phan
Dartmouth College

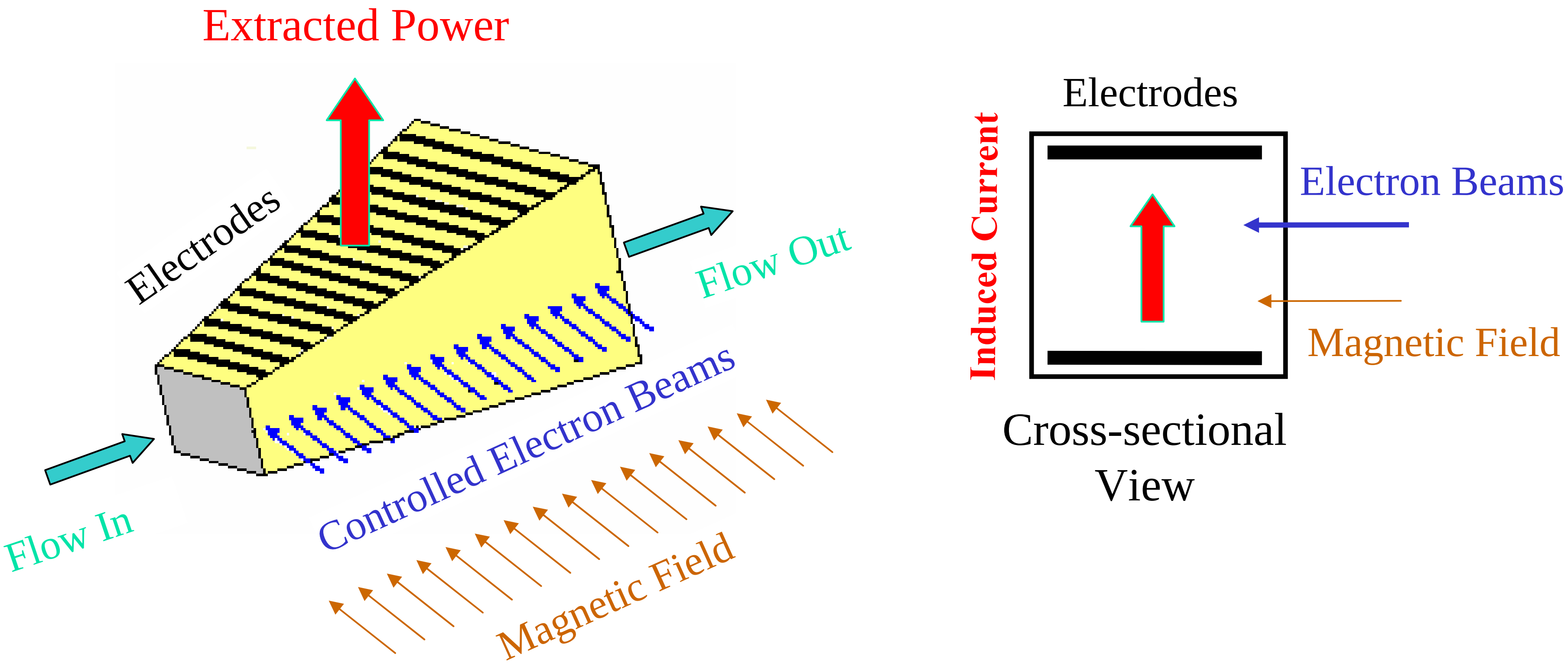
Prof. Robert F. Stengel
Princeton University

JUP Quarterly Review
October 23, 2003

Presentation Outline

- Hypersonic magneto-hydrodynamic (MHD) generator
- Dynamic programming based closed-loop optimal control approach review
- Reinforcement learning/adaptive critic based design
- Conclusions

Magneto-Hydrodynamic (MHD) Generator at the Inlet



Cross-sectional View

Schematic of the MHD Generator

MHD Generator Model

- Assumptions
  - One-dimensional steady state flow
  - Inviscid flow
  - No reactive chemistry
  - Low Magnetic Reynolds number
- *x*-*t* equivalence

Flow Equations

- Continuity Equation

$$\frac{d(\rho u A)}{dx} = 0$$

x - Coordinate along the channel

ρ - Fluid density

u - Fluid velocity

A - Channel cross-section area

- Force Equation

$$\rho u \frac{du}{dx} + \frac{dP}{dx} = -(1-k)\sigma u B^2$$

P - Fluid pressure

k - Load factor

σ - Fluid conductivity

B - Magnetic field

$$\sigma = \frac{\sigma_0}{(1+\Omega_e \Omega_i)}$$

Ω_e - Electron Hall parameter

Ω_i - Ion Hall parameter

Flow Equations…

- Energy Equation

$$\rho u \frac{d(\gamma\varepsilon + \frac{u^2}{2})}{dx} = -k(1-k)\sigma u^2 B^2 + Q_\beta$$

ε - Fluid internal energy

Q_β - Energy deposited by the e-beam

- Continuity Equation for the electron number density

$$\frac{d\left[\frac{n_e v}{1+(1-k)\Omega_e\Omega_i}\right]}{dx} = \frac{2 j_b \varepsilon_b}{eY_i Z} - \beta n_e^2 \varepsilon_b$$

n_e - Electron number density

j_b - Electron beam current

- E-beam energy

Z - Channel width

Y - Ionization potential

Performance Characterization

$$J = p_1\left[T(x_f) - T_e\right]^2 + p_2\left[M(x_f) - M_e\right]^2 + \\ + \int_0^{x_f} \left[\begin{array}{l} \dfrac{q_1}{\rho u A}\left[Q_\beta A - k(1-k)\sigma u^2 B^2 A\right] + \\ q_2 h(P) + q_3 dS^2 + r_1 j_b^2 \end{array}\right] dx$$

- Attaining prescribed values of flow variables at the channel exit (Mach number, Temperature)
- Maximizing the net energy extracted which is the difference between the energy extracted and the energy spent on the e-beam ionization
- Minimizing adverse pressure gradients
- Minimizing the entropy rise in the channel
- Minimizing the use of excessive electron beam current

Predictive Control Based Optimal Control

- Features of our optimal controller design technique
  - Works for both linear and nonlinear systems
  - Data-based
  - Finite horizon, end-point optimal control problem
  - Equivalent to time (position) varying system dynamics

[1] Kulkarni, N.V. and Phan, M.Q., "Data-Based Cost-To-Go Design for Optimal Control," *AIAA Paper* 2002-4668, *AIAA Guidance, Navigation and Control Conference*, August 2002.

[2] Kulkarni, N.V. and Phan, M.Q., "A Neural Networks Based Design of Optimal Controllers for Nonlinear Systems," *AIAA Paper* 2002-4664, *AIAA Guidance, Navigation and Control Conference*, August 2002.

Dynamic Programming Based State Feedback Control

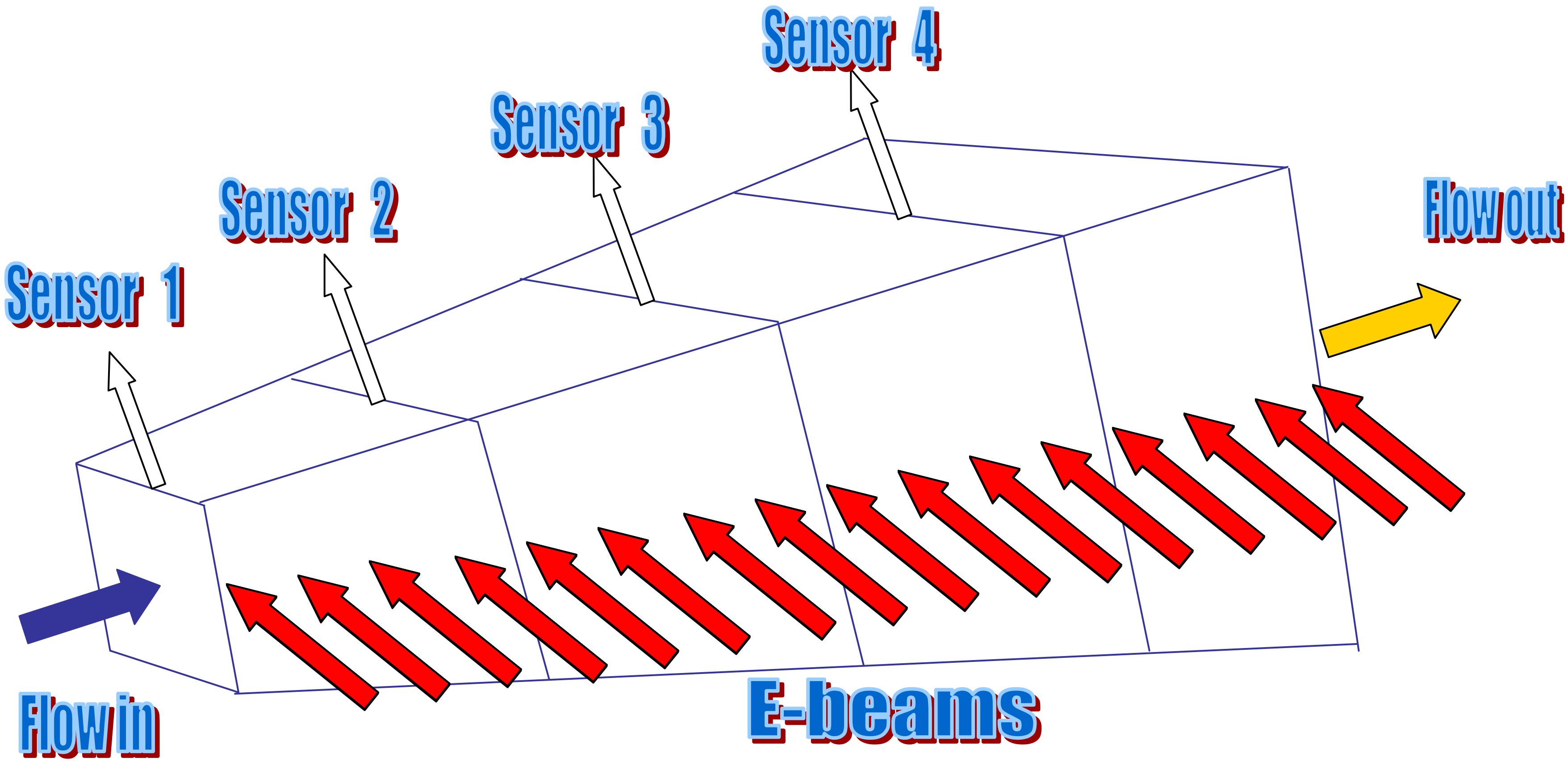
- Using the dynamic programming principle to design the controllers along the channel

  An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

  - Richard Bellman

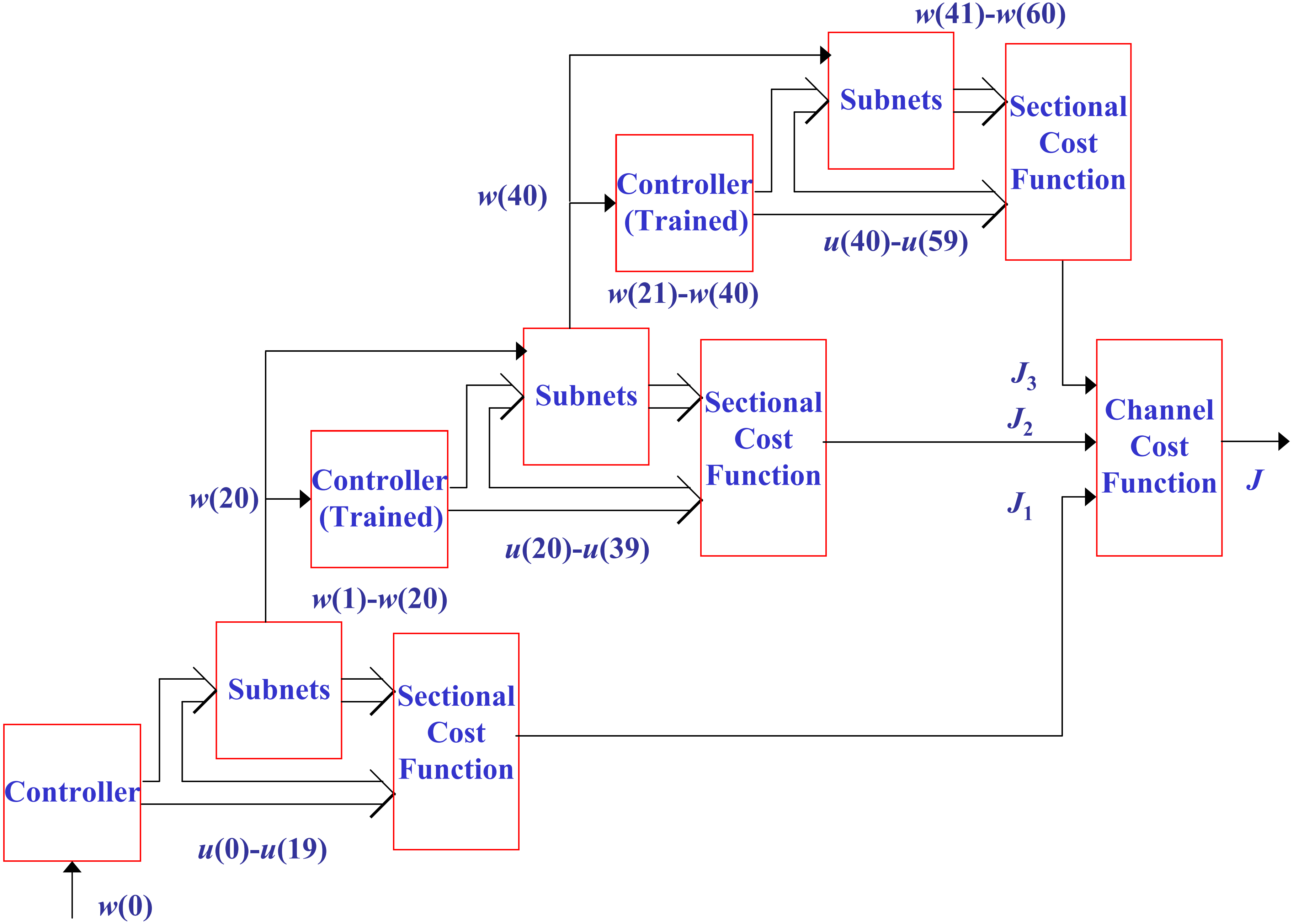
- Assume available sensors along the channel

Dynamic Programming Based State-Feedback Architecture

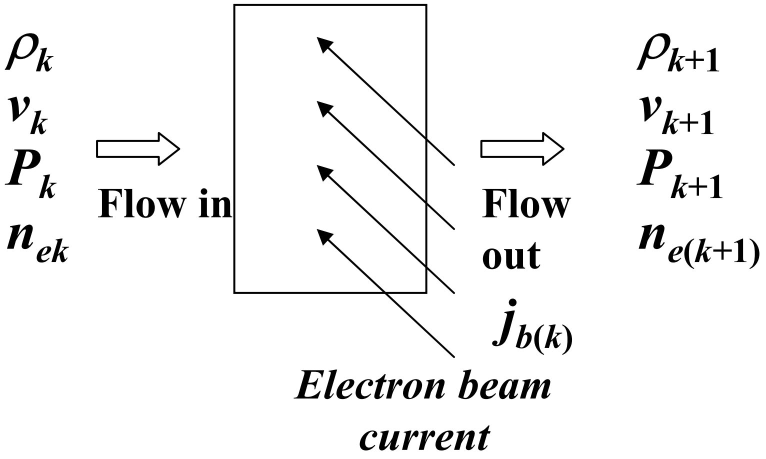


Inlet MHD Channel

Details of the Control Architecture

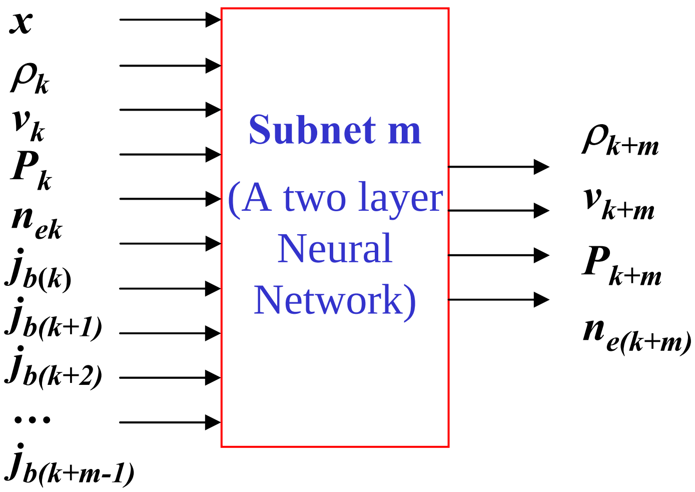


Using Subnets to Build the Cost Function Network



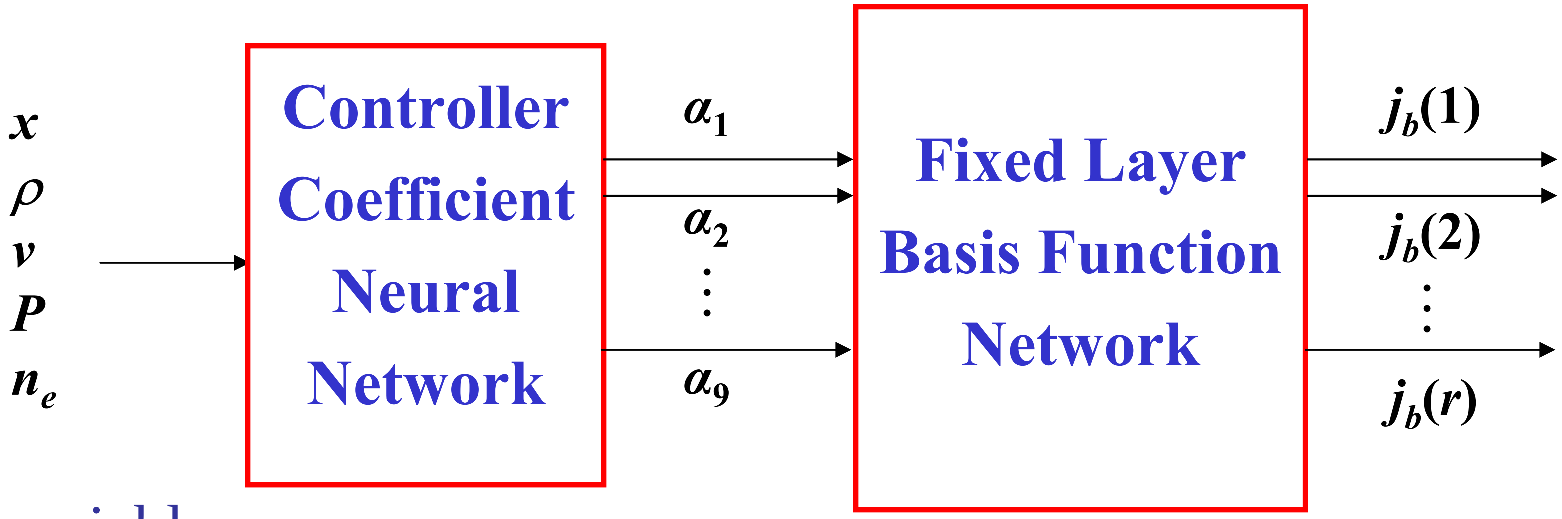
Physical picture describing Subnet 1

- Continuously spaced e-beam windows each having a length of 0.5 cm
- Subnet 1 chosen to correspond to the system dynamics between a group of 4 e-beam windows
- Length of the channel = 1 m
- Need subnets up to order 50



Subnet *m*, inputs and outputs.

Formulation of the Control Architecture: Neural Network Controller



Flow variables at the sensor location

Electron beam profile

$$j(i) = \sum_{k=1}^{9} \alpha_k \phi_k(i)$$

Reinforcement Learning/Adaptive Critic Architectures

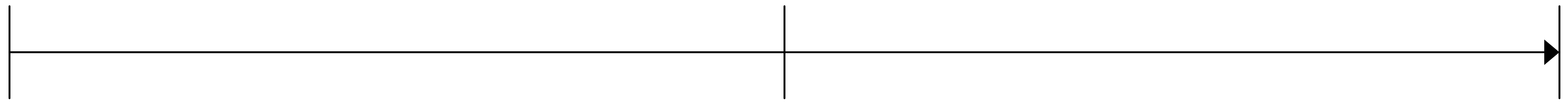
- Need to account for model uncertainties and disturbances or noise .
- Reinforcement Learning/Adaptive Critic Architectures are well-suited to handle model uncertainties and noise
- The existing design serves as a good starting point for controller/model updating
- The best features of the proposed dynamic programming based approach and the adaptive critic approaches can be combined

Value Function Based Reinforcement Learning

- Consider a system with a discrete state and action space
- Initial the cost-to-go function or the Value function for each state: $V(s_k)$
- Initialize the action policy (controller) for each state : $\pi(s_k)$
- Simulate an episode starting with a random initial state
- For each state occuring in the episode update the Value function using the principle of dynamic programming
- For each state occuring in the episode update the policy to minimize the value function
- Simulate another episode and loop

Implementing Value Function Based Reinforcement Learning

For a given episode:



$t = T - 2$ $t = T - 1$ $t = T$

Forward direction →

Given s_t
Choose $a_t = \pi(s_t)$
Get $r_t = r_t(s_t,a_t)$

Given s_t
Choose $a_t = \pi(s_t)$
Get $r_t = r_t(s_t,a_t)$

Get s_T
Get $r_T = r_T(s_T)$

← Backward direction

Update $\mathrm{V}(s_{T\text{-}2}) \leftarrow \mathrm{V}(s_{T\text{-}2}) + \alpha(\, r_{T\text{-}2} + V(s_{T\text{-}1}) - \mathrm{V}(s_{T\text{-}2})\,)$

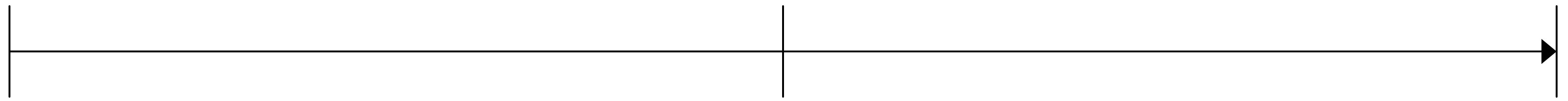
Update $\pi(s_{T\text{-}2}) = \arg\min_{a_t}\left(r_t(s_{T-2},a_t) + V_{t+1}(s_{t+1})\right)$

Update $\mathrm{V}(s_{T\text{-}1}) \leftarrow \mathrm{V}(s_{T\text{-}1}) + \alpha(\, r_{T\text{-}1} + r_T - \mathrm{V}(s_t)\,)$

Update $\pi(s_{T\text{-}1}) = \arg\min_{a_t}\left(r_t(s_{T-1},a_t) + r_T\right)$

Implementing Q-Function Based Reinforcement Learning

For a given episode:



$t = T - 2$ $t = T - 1$ $t = T$

Forward direction →

$t = T - 2$	$t = T - 1$	$t = T$
Given s_t Choose $a_t = \pi(s_t)$ Get $r_t = r_t(s_t,a_t)$	Given s_t Choose $a_t = \pi(s_t)$ Get $r_t = r_t(s_t,a_t)$	Get s_T Get $r_T = r_T(s_T)$

← Backward direction

Update $Q(s_{T-2}, a_{T-2}) \leftarrow Q(s_{T-2}, a_{T-2}) + \alpha(r_{T-1} + \arg\min_{a_t} Q_t(s_{T-1}, a_t) - Q(s_{T-2}, a_{T-2})))$

Update $Q(s_{T-1}, a_{T-1}) \leftarrow Q(s_{T-1}, a_{T-1}) + \alpha(r_T - Q(s_{T-1}, a_{T-1}))$

Update $\pi(s_{T-2}) = \arg\min_{a_t} Q_t(s_{T-2}, a_t)$

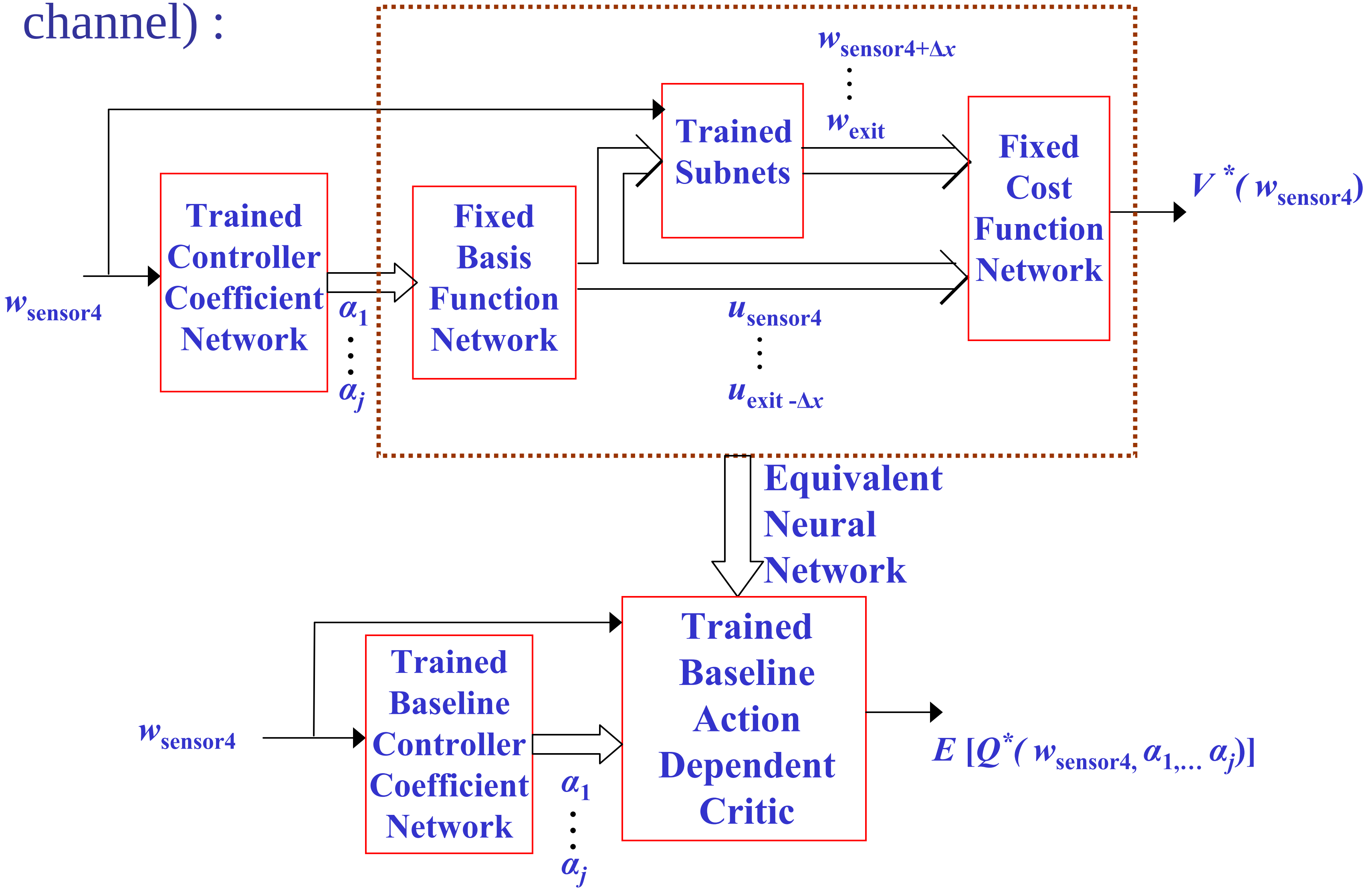
Update $\pi(s_{T-1}) = \arg\min_{a_t} Q_t(s_{T-1}, a_t)$

Q-Function Based Controller Updating the MHD Channel

- Develop the subnets from the available simulation package that captures the known physics
- Design the optimal controllers using the dynamic programming based control architecture.
- Obtain two layer network equivalence of the optimal controllers and the action dependent critics
- Provides good base line controllers & action dependent critics
- Improve these optimal controllers and the action dependent critics from real time data to account for stochastic factors (model uncertainties and disturbance or noise)

Implementation Details

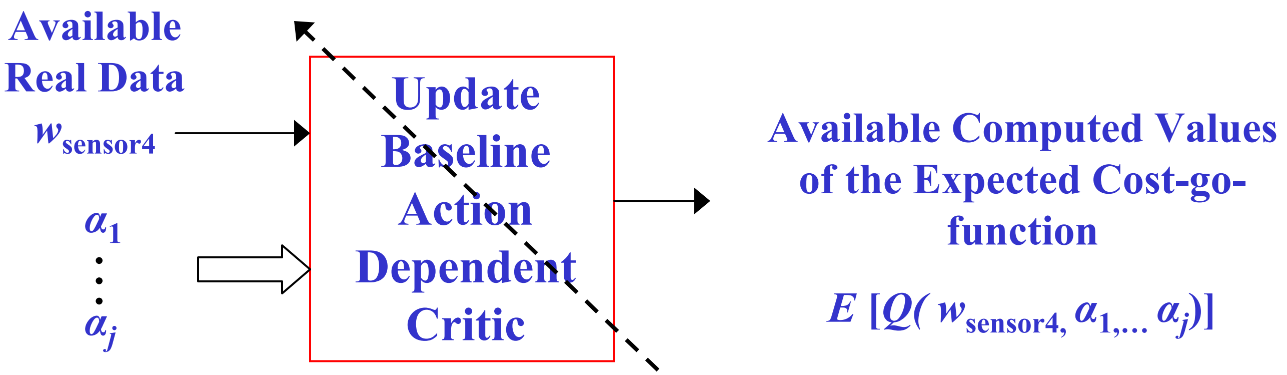
From sensor 4 to the end of the channel (Assume 4 sensors in the channel) :



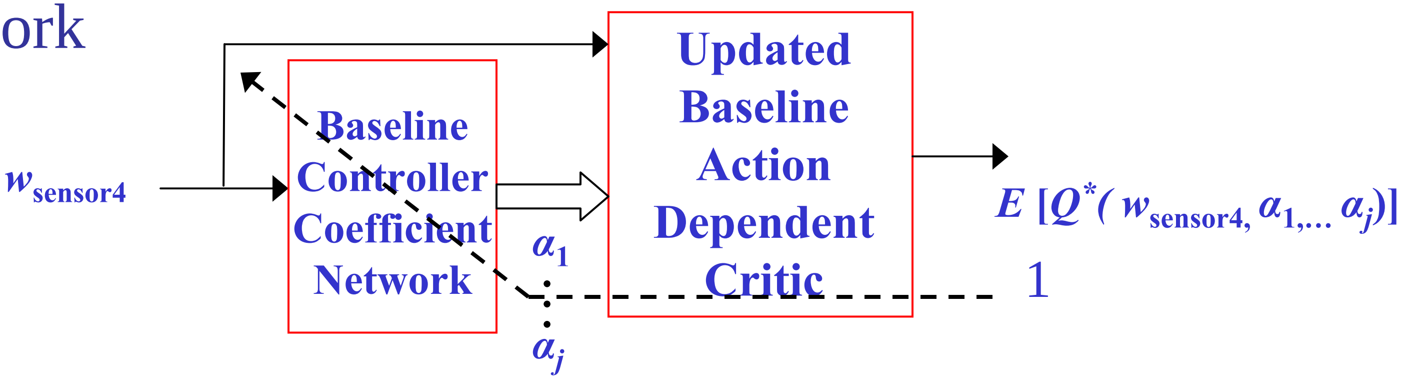
$w_{\text{sensor4}+\Delta x}$
$\vdots$
w_{exit}
Trained Subnets
Trained Controller Coefficient Network
w_{sensor4}
α_1
$\vdots$
α_j
Fixed Basis Function Network
Fixed Cost Function Network
$V^*(w_{\text{sensor4}})$
u_{sensor4}
$\vdots$
$u_{\text{exit}-\Delta x}$
Equivalent Neural Network
w_{sensor4}
Trained Baseline Controller Coefficient Network
α_1
$\vdots$
α_j
Trained Baseline Action Dependent Critic
$E\,[Q^*(w_{\text{sensor4}}, \alpha_1,\ldots \alpha_j)]$

Implementation Details

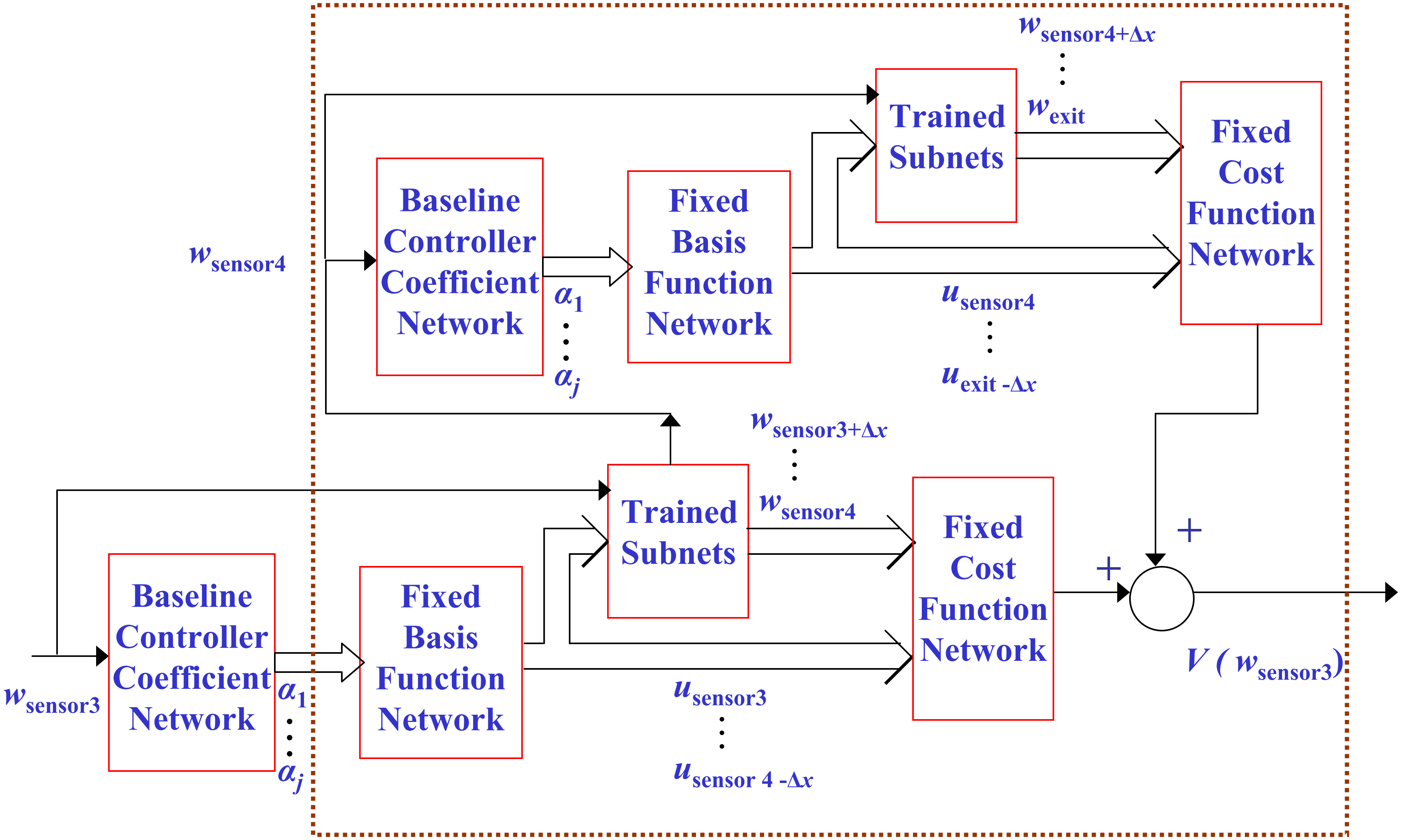
- With available real data calculate the new available values of expected cost-to-go function from sensor 4 to the end of the channel
- Update the baseline action dependent critic from sensor 4


Available Real Data
w_{sensor4}
α_1
α_j
Update Baseline Action Dependent Critic
Available Computed Values of the Expected Cost-go-function
$E\,[Q(\,w_{\text{sensor4},}\,\alpha_{1,\ldots}\,\alpha_j)]$

- With the updated ADC, update the baseline controller coefficient network

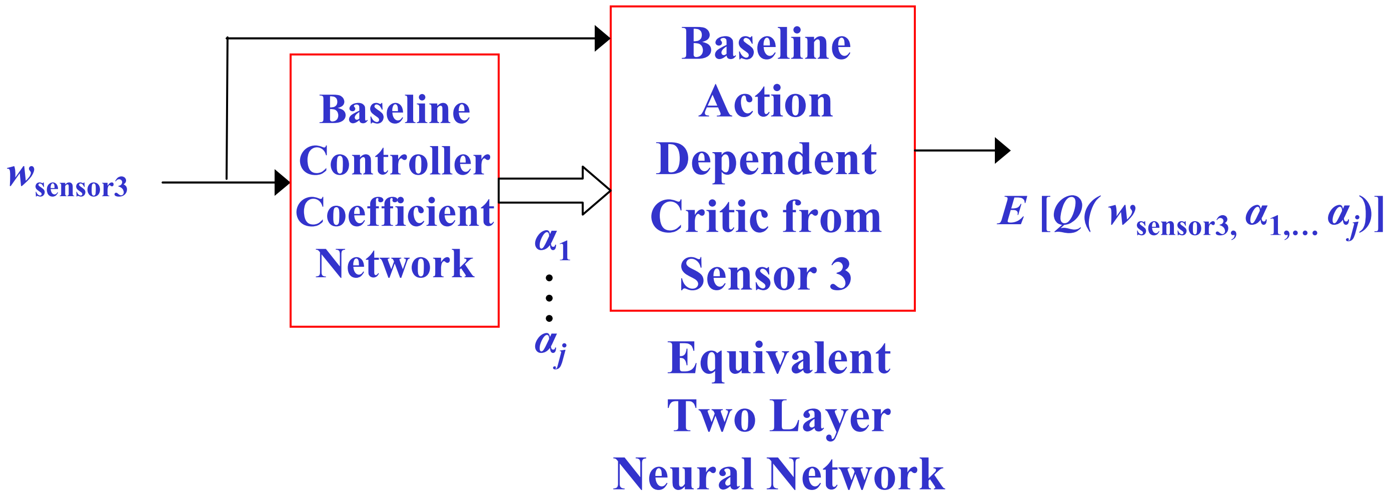

w_{sensor4}
Baseline Controller Coefficient Network
α_1
α_j
Updated Baseline Action Dependent Critic
$E\,[Q^*(\,w_{\text{sensor4},}\,\alpha_{1,\ldots}\,\alpha_j)]$
1

From sensor 3 to the end of the channel



Equivalent Action Dependent Critic from Sensor 3

From sensor 3 to the end of the channel



$w_{sensor3}$
Baseline Controller Coefficient Network
α_1
α_j
Baseline Action Dependent Critic from Sensor 3
$E\ [Q(w_{sensor3,}\ \alpha_{1,\dots}\ \alpha_j)]$
Equivalent Two Layer Neural Network

Action dependent critic-controller architecture from sensor 3

Updating the Sensor 3 ADC with Real Data

Critic Updating: Uses principle of optimality

$$E\ [Q(\ w_{\text{sensor3,}}\ \alpha_{1,\ldots}\ \alpha_j)]_{\text{desired}} = \mathrm{E}[U(\ w_{\text{sensor3,}}\ \alpha_{1,\ldots}\ \alpha_j)] + E\ [Q^*(\ w_{\text{sensor4,}}\ \alpha_{1,\ldots}\ \alpha_j)]$$

Used to update the baseline sensor 3 ADC

Available from the real data

Available from the updated sensor 4 ADC

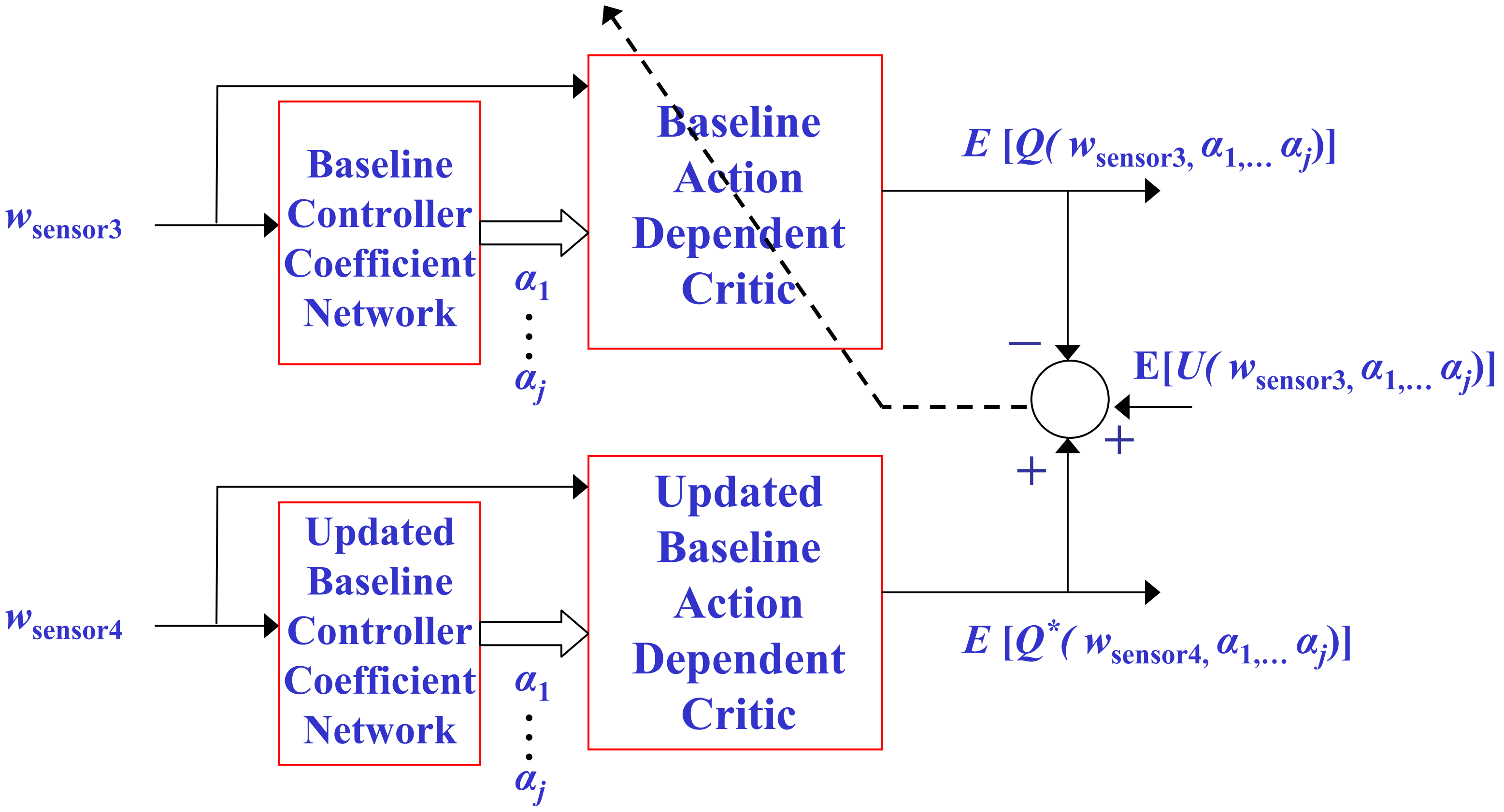
Controller Updating: Uses backpropagation

Backpropagate through the updated critic to get $\dfrac{\partial E[Q^*(w_{sensor3},\overline{\alpha})]}{\partial\overline{\alpha}}$

to update the controller

Critic-Controller Updating

Critic Updating:



w_{sensor3}
Baseline Controller Coefficient Network
α_1
α_j
Baseline Action Dependent Critic
$E\,[Q(\,w_{\text{sensor3}},\,\alpha_1,\ldots\,\alpha_j)]$
$-$
$E[U(\,w_{\text{sensor3}},\,\alpha_1,\ldots\,\alpha_j)]$
$+$
$+$
w_{sensor4}
Updated Baseline Controller Coefficient Network
α_1
α_j
Updated Baseline Action Dependent Critic
$E\,[Q^*(\,w_{\text{sensor4}},\,\alpha_1,\ldots\,\alpha_j)]$

Controller Updating:

Same as the controller updating from sensor 4

Conclusions

- Developed a neural network based optimal controller architecture for the hypersonic MHD channel
- Data-based approach
- Implemented open loop and closed loop designs
- ADHDP based design to account for stochastic factors in progress